SD

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/887,622	06/22/2001	Cary Lee Bates	ROC920010090US1	7424

7590 01/13/2005

Gero G. McClellan
Thomason, Moser & Patterson, L.L.P.
3040 Post Oak Boulevard, Suite 1500
Houston, TX 77056-6582

EXAMINER	
PHAM, CHRYSTINE	
ART UNIT	**PAPER NUMBER**
2122	

DATE MAILED: 01/13/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/887,622	BATES ET AL.	
	Examiner	Art Unit	
	Chrystine Pham	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *1 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *01 October 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the amendment filed on October 1st 2004. Claims 1-42 are presented for examination.

Response to Amendments

2. In view of Applicants' amendments to the specification in response to the objection to the specification containing inconsistency in references made to drawings, the objection to the specification is hereby withdrawn.

3. In view of Applicants' amendments to drawings in response to the objection to the drawings lacking clear indication of connections between related elements, the objection to the drawings is hereby withdrawn.

Response to Arguments

4. Applicants' arguments filed on October 1st 2004 in regards to claim rejections have been fully considered but they are not persuasive.

Per claim 1, the Applicants essentially contend that the cited portions (col.23:51-60) of Sumi et al. "only refer to an 'operation-possible variable display unit' which provides an indication of variables that may be referenced or set via a user command input via a command line (see col.23, lines 25-41) by determining if such variables have been allocated resources ... variables that maybe viewed and/or modified by the user via the debugger are identified, not variables that will be used and/or changed by subsequent program execution, as claimed" (page 12 last paragraph and continues on page 13 of paper dated October 1st 2004). The examiner respectfully disagrees. The cited portions of Sumi et al. discuss an operation-possible variable display unit, as illustrated in FIG.9A (see *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text) which clearly shows that the teaching of Sumi et al. is not limited to showing the executable status of variables which may be referenced (that is to say,

used) or set (that is to say, changed) via a user command line, as submitted by Applicants. Referring to FIG.9A, program execution halts at Line 10 in Line Information Display Window, where variable z is being changed (hence used) from the integer value of 10 to new integer value of 20 (see z column of *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* at Line 10 FIG.9A; see *OUTPUT WINDOW* FIG.9A & associated text). While program execution is still at a halt, the Operation-possible variable display window displays a circle in the x column (at Line 10) **visibly indicating the executable status of least one variable** (i.e., variable x), **wherein the executable status is indicative of at least one of a use and change** (see *x 20* in *VARIABLE AUTOMATIC DISPLAY WINDOW* FIG.9A; see at least last 2 lines of *OUTPUT WINDOW* FIG.9A) **of the current value** (see *x=getData();* on Line 6 of *LINE DISPLAY WINDOW* FIG.9A) **during subsequent program execution** (see circle in x column on Line 11 of *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text), as claimed. Thus, the change in value of variable x does not require the user command input via command line as submitted by Applicants. Furthermore, the cited portions (col.23:51-60) discusses the steps of FIG.12 (see *S43-S49* FIG.12 & associated text) which clearly shows that the at least one variable will be used and/or changed by subsequent program execution as *INVETIGATION OBJECT LINE* moves to the next executable statement (see *S48* FIG.12 & associated text) in addition to the discussion of FIG.9A above.

Per claims 6, 17, and 29, the Applicants submitted that the cited portions of Sumi et al. do not teach "... determining at least one of a first executable status and a second executable status of the at least one variable based on a current point of execution ..." and "... visually indicates an executable status of the at least one variable at the current point of execution" (page 13 first full paragraph of paper dated October 1st 2004). The examiner respectfully disagrees. As discussed above in regards to FIG.9A, it has been shown that the at least one of a first executable status (see z column Line 10 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text) and a second executable status of the at least one variable is determined and visually indicated at a current point of execution (see *HALT POSITION, Line 10* FIG.9A &

associated text), as claimed. The Applicants further submitted that the cited portions "rather teach different types of operations, such as determining whether variables have been replaced during optimization operations or whether resources have been allocated to a variable .." (page 13 first full paragraph of paper dated October 1st 2004). The examiner respectfully submits that these types of operations, as categorized by the Applicants, clearly teach "determining the at least one of a first executable status and a second executable status of the at least one variable" since determining, and indicating the replaceable status of a variable in subsequent program execution, as discussed above in regards to FIG.9A and FIG.12, is equivalent to determining, and visually indicating the first executable status, which is defined by whether the current value of the variable may change during subsequent execution, and determining, and indicating "whether resources have been allocated to a variable" is equivalent to determining, and indicating the second executable status, which is defined by the use of the variable in subsequent program execution, as shown above in discussion of FIG.9A, and FIG.12.

In view of the foregoing discussion, the examiner considers that the rejection of the claims under 35 U.S.C 102(b) and 103(a) is proper and maintained.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –
(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-8, 10-15, 17-24, 27, 29-30, 32-34, and 37-41 are rejected under 35 U.S.C. 102(b) as being anticipated by Sumi et al. (U.S. Patent 5,881,288) (hereinafter *Sumi et al.*).

As per claim 1, *Sumi et al.* disclose a computer system, comprising an output device (e.g., FIG.4 *502* & associated text) and at least one processor (e.g., FIG.7 & associated text) which, when executing a debugging program, is configured to:

- wait for a program being debugged to stop executing immediately prior to executing a next executable statement (e.g., see Line 10 in *Line Information Display Window* FIG.9A & associated text) at which at least one variable has a current value (e.g.,col.24 : 23-27 & col.27 : 48-52; see z, x columns Line 10 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text;); and
- display on the output device the at least one variable in a manner that visually indicates an executable status of the at least one variable, wherein the executable status is indicative of at least one of a use and change of the current value (e.g., see *x=getData();* in *LINE DISPLAY WINDOW* FIG.9A & associated text) during subsequent continuing execution of the program being debugged (e.g., see x column Line 11 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text; see at least last 2 lines in *OUTPUT WINDOW* FIG.9A & associated text; see *x 20* in *VARIABLE AUTOMATIC DISPLAY WINDOW* FIG.9A & associated text; col.23 : 51-60; see *S43-S49* FIG.12 & associated text).

As per claim 2, *Sumi et al.* disclose a system as applied to claim 1, wherein the executable status indicates that the current value may change when the next executable statement is executed (e.g., col.23 : 31-36).

As per claim 3, *Sumi et al.* disclose a system as applied to claim 1, wherein the executable status indicates that the current value may be used when the next executable statement is executed (e.g., col.27 : 39-41).

As per claim 4, *Sumi et al.* disclose a system as applied to claim 1, wherein the executable status is visually represented on the output device to differentiate the at least one variable from other variables displayed on the output device (e.g., FIG.9A *operation-possible variable display window* & FIG.9B).

As per claim 5, *Sumi et al.* disclose a system as applied to claim 1, further comprising a memory containing a monitor window interface (e.g., FIG.4 *214*) configured to display the at least one variable on the output device in a manner to visually differentiate the at least one variable from other variables having a different executable status (e.g., FIG.9A *operation-possible variable display window* & FIG.9A & associated text).

As per claim 6, *Sumi et al.* disclose a method for displaying variables of a program being debugged, comprising:

- when the program being debugged stops executing immediately prior to executing a next executable statement at which at least one variable has a current value (e.g., col.24 : 23-27 & col.27 : 48-52; see *HALT POSITION, Line 10* FIG.9A & associated text), determining at least one of a first executable status and a second executable status of the at least one variable based on a current point of execution (e.g., see z column Line 10 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text), wherein the first executable status is defined by whether the current value of the at least one variable may change during subsequent execution of the program being debugged (e.g., see x column on Line 11 of *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text) and the second executable status is defined by whether the current value of the at least one variable has a use during subsequent execution of the program being debugged (e.g., FIG.18A *S76* & *S79*, col.6 : 56-62, col. 11 : 47-52, col.15 : 27-36, col.18 : 25-35, col.23 : 25-41 and 51-60, and col.27 : 30-36); and

- preparing an output which, when displayed on an output device, visually indicates the executable status of the at least one variable at the current point of execution (e.g., FIG.6A & FIG.5D *primitive storage unit* & associated text; see x, z columns in Line 10 and x column in Line 11 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text; col.15 : 32-49; see FIG.12 & associated text).

As per claim 7, *Sumi et al.* disclose a method as applied to claim 6, wherein the second executable status is defined according to one of only the next executable statement and any statement that may be encountered during subsequent execution of the program being debugged (e.g., col.18 : 26-35).

As per claim 8, *Sumi et al.* disclose a method as applied to claim 6, wherein the program being debugged stops executing upon encountering a breakpoint (e.g., col.19 : 25-42, and col.21 : 38-40).

As per claim 10, *Sumi et al.* disclose a method of claim 6, wherein preparing comprises preparing the output so that, when displayed, the at least one variable is visually differentiable from other displayed variables according to their respective executable statuses (e.g., FIG.6A & FIG.5D *primitive storage unit* & associated text, FIG.9A, and col.15 : 32-49).

As per claim 11, *Sumi et al.* disclose a method of claim 6, wherein the first executable status indicates that the value of the at least one variable may change during subsequent execution of the program being debugged (e.g., FIG.12 *S43-S49* & associated text).

As per claim 12, *Sumi et al.* disclose a method of claim 6, wherein the first executable status indicates that the value of the at least one variable may change during subsequent execution of the program being debugged and the second executable status indicates that the

value of the at least one variable has a use during subsequent execution of the program being debugged (e.g., FIG.12 *S43-S49* & associated text, col.27 : 30-43).

As per claim 13, *Sumi et al.* disclose a method of claim 6, wherein at least one variable is a variable referenced in the next executable statement in the program being debugged (e.g., col.15 : 26-28, FIG.9A variable ***x*** on line 10 & 11 of *line information display window, line display window, and operation-possible variable display window*).

As per claim 14, *Sumi et al.* disclose a method of claim 13, wherein the next executable statement contains a plurality of variables (e.g., FIG.12 *S48* & *S49* & associated text), and wherein preparing comprises preparing the output so that, when displayed, the at least one variable and the plurality of variables are visually differentiable from one another according to their respective different executable statuses (e.g., FIG.6A & FIG.5D *primitive storage unit* & associated text, FIG.9A, and col.15 : 32-49, col.23 : 25-32).

As per claim 15, it recites limitation, which has been addressed in claim 14 above, therefore, is rejected for the same reason as cited in claim 14.

As per claim 17, *Sumi et al.* disclose a method for displaying variables of a program being debugged, comprising:

- when a program being debugged stops executing immediately prior to a next executable statement at which at least one variable has a current value (e.g., col.24 : 23-25, and col.27 : 48-52; see *HALT POSITION, Line 10* FIG.9A & associated text), determining an executable status of at least one variable of the statement based on a current point of execution (e.g., see z column Line 10 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text), wherein the executable status is indicative of at least one of a possible use and a possible change of the current value during

subsequent continuing execution of the program being debugged (e.g., FIG.18A *S76* & *S79*, col.6 : 56-62, col. 11 : 47-52, col.15 : 27-36, col.18 : 25-35, col.23 : 25-41 and 51-60, and col.27 : 30-36; see x column on Line 11 of *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text); and

- preparing an output which, when displayed on an output device, visually indicates the executable status of the at least one variable at the current point of execution (e.g., FIG.6A & FIG.5D *primitive storage unit* & associated text, FIG.9A, and col.15 : 32-49, col.23 : 25-32; see x, z columns in Line 10 and x column in Line 11 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text; see FIG.12 & associated text).

As per claim 18, *Sumi et al.* disclose a method of claim 17, wherein the executable status is defined according to one of only the next executable statement and any statement that may be encountered during the subsequent continuing execution of the program being debugged (e.g., FIG.12 *S43-S49* & associated text, col.18 : 26-34, and col.24 :5-8).

As per claim 19, *Sumi et al.* disclose a method of claim 17, wherein the program being debugged stops executing upon encountering a breakpoint (e.g., col.19 : 25-42, and col.21 : 38-40).

As per claims 20-22, they recite limitations, which have been addressed in the above claims 12, 10, and 14 (respectively), therefore, are rejected for the same reasons as cited in claims 12,10, and 14 respectively.

As per claims 23-24, they recite limitations, which have been addressed in both of the above claims 12 & 10, therefore, are rejected for the same reasons as cited in both claims 12 & 10.

As per claim 27, it recites limitation, which has been addressed in the above claim 10, therefore, is rejected for the same reason as cited in claim 10.

As per claim 29, *Sumi et al.* disclose a signal bearing medium (e.g., FIG.4), comprising a debugging program which, when executed by a processor, performs a method, comprising:

- when a program being debugged stops executing immediately prior to a next executable statement at which at least one variable has a current value (e.g., col.24 : 23-27 & col.27 : 48-52; see *HALT POSITION, Line 10* FIG.9A & associated text), determining an executable status of at least one variable of the statement based on a current point of execution (e.g., see z column Line 10 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text), wherein the executable status is indicative of at least one of a possible use and a possible change of the current value during subsequent continuing execution of the program being debugged (e.g., FIG.18A *S76* & *S79*, col.6 : 56-62, col. 11 : 47-52, col.15 : 27-36, col.18 : 25-35, col.23 : 25-41 and 51-60, and col.27 : 30-36; see x column on Line 11 of *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text); and
- preparing an output which, when displayed on an output device, visually indicates the executable status of the at least one variable at the current point of execution (e.g., FIG.6A & FIG.5D *primitive storage unit* & associated text, FIG.9A, and col.15 : 32-49, col.23 : 25-32; see x, z columns in Line 10 and x column in Line 11 in *OPERATION-POSSIBLE VARIABLE DISPLAY WINDOW* FIG.9A & associated text; see FIG.12 & associated text).

As per claims 30 and 32, they recite limitations which have been addressed in the above claims 8, and 7 respectively, therefore, are rejected for the same reasons as cited in claims 8, and 7.

As per claims 33-34, 37-39, and 41, they recite limitations, which have been addressed in the above claims 10, and 12, therefore, are rejected for the same reasons as cited in claims 10 & 12.

As per claim 40, it recites limitations, which have been addressed in the above claim 22, therefore, is rejected for the same reasons as cited in claim 22.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 9, 25-26, 31, 35-36, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Sumi et al.* as applied to claims 6, 17, and 29 above, and further in view of Miyata et al. (U.S. Patent 5,165,036) (hereinafter *Miyata et al.*).

As per claim 9, *Sumi et al.* teach a method as applied to the above claim 6. Sumi et al. fail to teach determining which comprises referring to a control flow graph (CFG). However, *Miyata et al.* disclose a method for displaying variables of a program being debugged, wherein determining comprises referring to a CFG (e.g., FIG.18, FIG.20, see *data flow program* col.4 : 22-27). It would have been obvious that one of ordinary skill in the pertinent art at the time of applicant's invention would be motivated to modify the teaching of *Sumi et al.* to include the use of a CFG as disclosed by *Miyata et al.*, since information contained in the CFG can be used by the debugging program to automatically

identify the decision points in the debugged program, whereat breakpoints are set, automatically.

As per claim 25, *Sumi et al.* teaching as modified by *Miyata et al.* (see above claim 9) teach a method as applied to claim 17, wherein determining comprises accessing a variable-containing data structure associated with the next executable statement (e.g., FIG.4 *function information storage unit 1042* & associated text, col.23 : 42-61).

As per claims 26, 31, and 35, they recite limitations which have been addressed in the above claim 25, therefore, are rejected for the same reasons as cited in claim 25.

As per claim 36, *Sumi et al.* teaching as modified by *Miyata et al.* (see above claim 9) teach the signal bearing medium of claim 35, wherein preparing comprises preparing the output so that, when displayed, the variables are visually differentiable from one another according to their respective executable statuses (see above claim 10).

9. Claims 16, 28, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over *Sumi et al.* as applied to claims 6, 17, and 29 (respectively) above, and further in view of Bates et al. (U.S. Patent 6,658,649) (hereinafter *Bates et al.*).

As per claim 16, *Sumi et al.* teach a method as applied to above claim 15. *Sumi et al.* fails to teach the formatting comprises at least one of brackets, parentheses, asterisks, highlighting, strike-outs and numerals. However, *Bates et al.* disclose a method for displaying variables (e.g., col.5 : 27-30), a step region which can be formatted by highlighting, shading, coloring, and the like (e.g., col.7 : 23-29), and the executed statements contained within the step region which can be formatted using asterisks,

highlighting, etc. (e.g., col.7 : 43-51). Even though, *Bates et al.* are not explicit, in particular, on the formatting of variables, it would have been obvious that the teaching of *Bates et al.* could have been easily modified to include the formatting of variables which comprise at least one of brackets, parentheses, asterisks, highlighting, strike-outs and numerals. In addition, it would have been obvious to one of ordinary skill in the pertinent art at the time of applicant's invention to modify the teaching of *Sumi et al.* to include the display formatting as disclosed by *Bates et al.*, since such display formats would further enhance visual distinctness of variables and further support their identification during debugging.

As per claims 28 and 42, they recite limitations which have been addressed in the above claim 16, therefore, are rejected for the same reasons as cited in claim 16.

Conclusion

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chrystine Pham whose telephone number is 571-212-3702. The examiner can normally be reached on Mon-Fri, 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

January 7, 2005

TUAN DAM
SUPERVISORY PATENT EXAMINER